Calculate the throughput in two-lane Traffic Cellular Automaton

Name: Yiqun Zhang **Mentor Name: Gray Lawrence**

School of Mathematics, College of Science and Engineering, University of Minnesota

Introduction

Cellular Automaton (CA) is basically a particle system that can be used to simulate traffic in a discrete system, which is so-called Traffic Cellular Automaton (TCA). A two-lane TCA system consists of two rows of cells in which each cell can have at most one particle (one car). The particle can move from one cell to the one that is the closest to its right, and it is assumed that the particle can only move forward (to the right). If it is blocked ahead by another particle, it will change lanes.

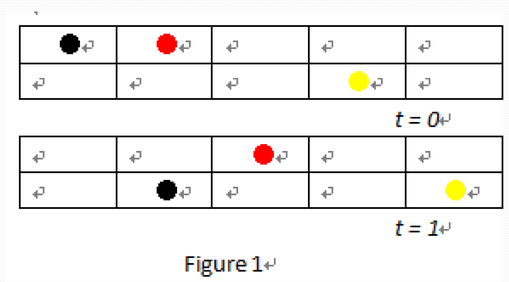


Figure 1

Figure 2.

It is necessary to clarify the rules that particles follow in 2-lane system. In figure 1, the red and yellow particle are moving forward since there are no particles ahead of them, and the black particle changes lanes because the red particle blocks it. In figure 2, the black particle is not able to change lanes if either one of the white dots is occupied by a particle, and hence it cannot move in this situation.

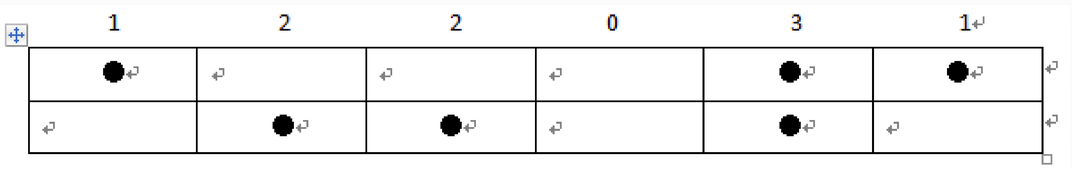
Experimental Method

Mathematica is a good software that could be used to simulate TCA because it has some built-in functions that are essential to construct our model.

Result

The CellularAutomaton function in Mathematica works well for a TCA with only one lane of traffic and no randomness. A major part of my project has been to find a way for it to work for two-lane traffic with randomness.

In order to handle two lanes, I reduce the two lanes of traffic into one lane by using a code of four numbers, 0,1,2,3. When one of these numbers appears in a single-lane cell, it indicates a situation in the two-lane model. Here is an example to illustrate this idea.



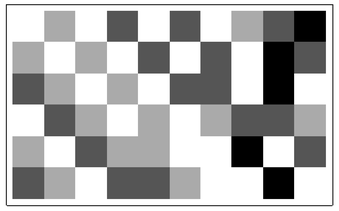
The screenshot above gives a state {1,2,2,0,3,1}. A function “f1[ ]” is used to define the rule, under which the particle will move and change lanes with probability m. In order to handle randomness, I added a decimal part to each number. For example, 1.35 stands for a 1, together with a probability value of .35 that can be used to determine whether the car represented by the 1 moves in the next time step. But when these values are displayed in the output plot, the decimal parts are removed.

```
If[1 + m < x[[2]] < 2 && 2 < x[[3]] < 2 + m && 0 < x[[4]] < 1 || 1 + m < x[[2]] < 2 && 2 < x[[3]] < 2 + m && 1 < x[[4]] < 2 ||
  1 + m < x[[2]] < 2 && 2 < x[[3]] < 2 + m && 2 < x[[4]] < 3, 0 * (++k), 0] +
If[1 < x[[2]] < 1 + m && 2 + m < x[[3]] < 3 && 0 < x[[4]] < 1 || 1 < x[[2]] < 1 + m && 2 + m < x[[3]] < 3 && 1 < x[[4]] < 2, 3, 0] +
If[1 < x[[2]] < 1 + m && 2 < x[[3]] < 2 + m && 0 < x[[4]] < 1 || 1 < x[[2]] < 1 + m && 2 < x[[3]] < 2 + m && 1 < x[[4]] < 2, 1 + 0 * (++k), 0] +
If[1 + m < x[[2]] < 2 && 2 + m < x[[3]] < 3 && 0 < x[[4]] < 1 || 1 + m < x[[2]] < 2 && 2 + m < x[[3]] < 3 && 1 < x[[4]] < 2, 2, 0] +
```

The screenshot above is a small part of my function f1[ ] in which parameter k is the counter that counts total number of movements of particles. It might takes several sentences to explain my function but I can explain it to you if you are interested in.

```
In[60]:= ArrayPlot[
           IntegerPart[CellularAutomaton[{(f1[#] + RandomReal[]) &, {}, 2}, RandomInteger[{1, 3}, size] * UnitStep[RandomReal[{-1 + p, p}, size]] + RandomReal[1, size],
             t]]]
```

Out[60]=



```
In[61]:= k / (size * t) // N
Out[61]= 0.54
```

After taking the integer part of numbers and making an arrayplot, there are “light gray”, “dark gray” and “black” colors which represent number “1”, “2”, and “3”. It can be observed that particles are moving and lane-changing, and each row represent the state of a particular time step. k/size/t is used to denote the throughput (the number of particles that passes through a particular cell per unit of time) of the system.

Reference

Gray, Lawrence and Griffeath, David, “ The Ergodic Theory of Traffic Jams”, Journal of Statistical Physics, Vol. 105, Nos. 3/4, pp. 415-418 November 2001.

Acknowledgement

Thanks Professor Gray Lawrence and Undergraduate Research Opportunity Program (UROP) for giving me this opportunity to present this research.